



Project Documentation

Class: ISYS 567 – Internship

Instructor: Prof. Verma

Students: Brandon Lai
Pascal Schuele



Table of Contents

1.) Introduction to Cloud Computing	3
2.) Public vs. Private Cloud.....	3
3.) Open source software for private clouds	4
4.) Openstack as example of open source software for private clouds.....	5
5.) How-To set up a private cloud using Openstack	6
5.1) SSH connection to the server	6
5.2) Downloading devstack.....	7
5.3) Execution of the script “stack.sh”	8
5.4) Log In to Openstack.....	9
5.5) Overview	10
5.6) Instances	11
5.7) Volumes.....	12
5.8) Flavors	13
5.9) Images	14
5.10) Projects.....	15
5.11) Users	16
5.12) System Info.....	17
5.13) Adding an image to Openstack.....	18

1.) Introduction to Cloud Computing

Cloud computing is the new frontier for technology, with the ability to scale from home use to enterprise use in the blink of an eye. Cloud computing, at its simplest, is just services that are being delivered to the user by use of a network (usually the Internet). There are two main services that cloud services provide, compute and storage. With the compute cloud comes a flurry of other services, such as Software as a Service (SaaS), Platform as a Service (PaaS), Desktop as a Service (DaaS), Security as a Service (SECaaS) and more. The storage service of cloud just acts like folder, which just holds the user's data. This provides a centralized system for the files. These services are versatile and most people feel it is more secure to run them on-site. In the cloud computing business, there are two main types of clouds, public and private.

In our project we will explain the main differences between these different types of clouds. We will take a quick look through the different options for private cloud software. After that, we will single out one up and coming open source software (Openstack) to implement a private cloud. Finally we provide a How-To with all the instructions needed, to set up a private cloud with Openstack.

2.) Public vs. Private Cloud

Public clouds are cloud systems that are available for everyone's use, such as Dropbox and Amazon's EC2. These services can be both free and subscription based, depending on the user's needs. A private cloud can offer the same services as a public cloud, however, its services are limited to people behind the company's firewall. There are a few reasons to choose a private cloud over a public cloud, and these options will be explored here. There are a couple main reasons why cloud computing has been on the rise. One reason is elasticity and scalability. "This encompasses the idea of computing on demand, and the ability to increase the supply of computing resources as they are needed to deal with spikes in demand for a particular application or service. There's also the idea of turning computing resources into a commodity so more can be

added over time, as needed, to ensure systems are almost infinitely scalable.”¹ This allows variable amounts of users on at the same time, or even enabled whenever it is necessary to alleviate bottlenecks.

There is a downside to private clouds, cost. The capital of hosting a private cloud is substantially more than having a public cloud hosted. The need to purchase the hardware, as well as hiring an administrator to manage the private cloud is the main downsides of it.

In general, public clouds are highly structured and automated. Usually it's not possible for enterprises to get particular SLA's (Service Level Agreement) for their specific needs. Public clouds usually come with a standard SLA, so you just have the choice to take it or leave it. This is one important fact, any organization which thinks about using public clouds, should always keep in mind: You have to adapt to the cloud, the cloud doesn't adapt to you. Basically you should first think about how critical the data is which you want to store in a cloud. The more critical the data is, the more important is it to keep them in house.

3.) Open source software for private clouds

Open source software for private clouds is widely available. Openstack, Eucalyptus, Ganeti, OpenNebula are the main competitors in the private cloud area. All of these except for Ganeti provide a clean user GUI, so we will just compare Openstack, Eucalyptus and OpenNebula. Out of the 3 remaining, OpenNebula focuses tremendously on Data Center virtualization, and has fewer options for controlling what we want from a private cloud. Therefore, we are left with Eucalyptus and Openstack. With Eucalyptus, there are many tools that are comparable to Openstack's, however, Openstack has a more user-friendly GUI and the functions seem simpler to use and deploy. After taking a look at the available software, Openstack fits our needs best.

¹ P. Rubens, January 27, 2010 - <http://www.serverwatch.com/trends/article.php/3861191/Private-Cloud-Defined.htm>

4.) Openstack as example of open source software for private clouds

“Openstack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.”² Openstack has many features that make deploying and managing a private cloud simple. “Openstack is architected to provide flexibility as you design your cloud, with no proprietary hardware or software requirements and the ability to integrate with legacy systems and third party technologies. It is designed to manage and automate pools of compute resources and can work with widely available virtualization technologies, as well as bare metal and high-performance computing (HPC) configurations”³. We are able to run Openstack on top of Ubuntu. Openstack has the capabilities to make the transition from other cloud services with load balancing, which allows it to migrate the services without downtime to the end users. Since Openstack has met the criteria for this project, we will move forward and install it on our test server running Ubuntu 10.4.

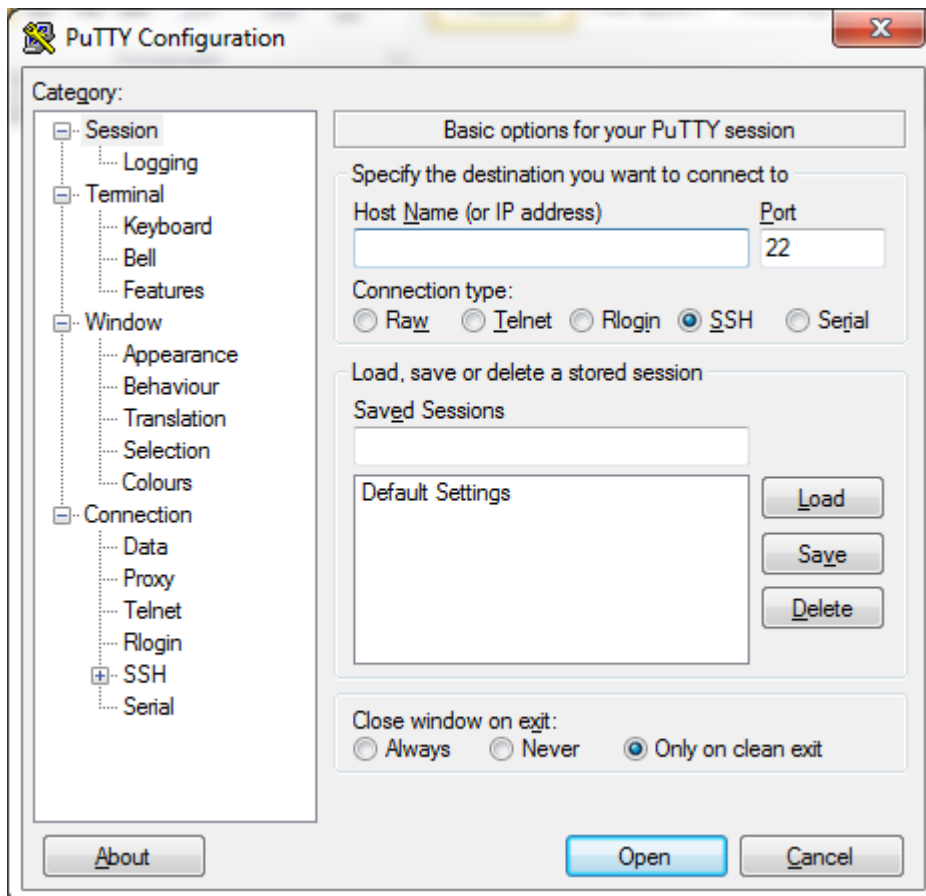
² <http://www.Openstack.org/software/>

³ <http://www.Openstack.org/software/Openstack-compute/>

5.) How-To set up a private cloud using Openstack

5.1) SSH connection to the server

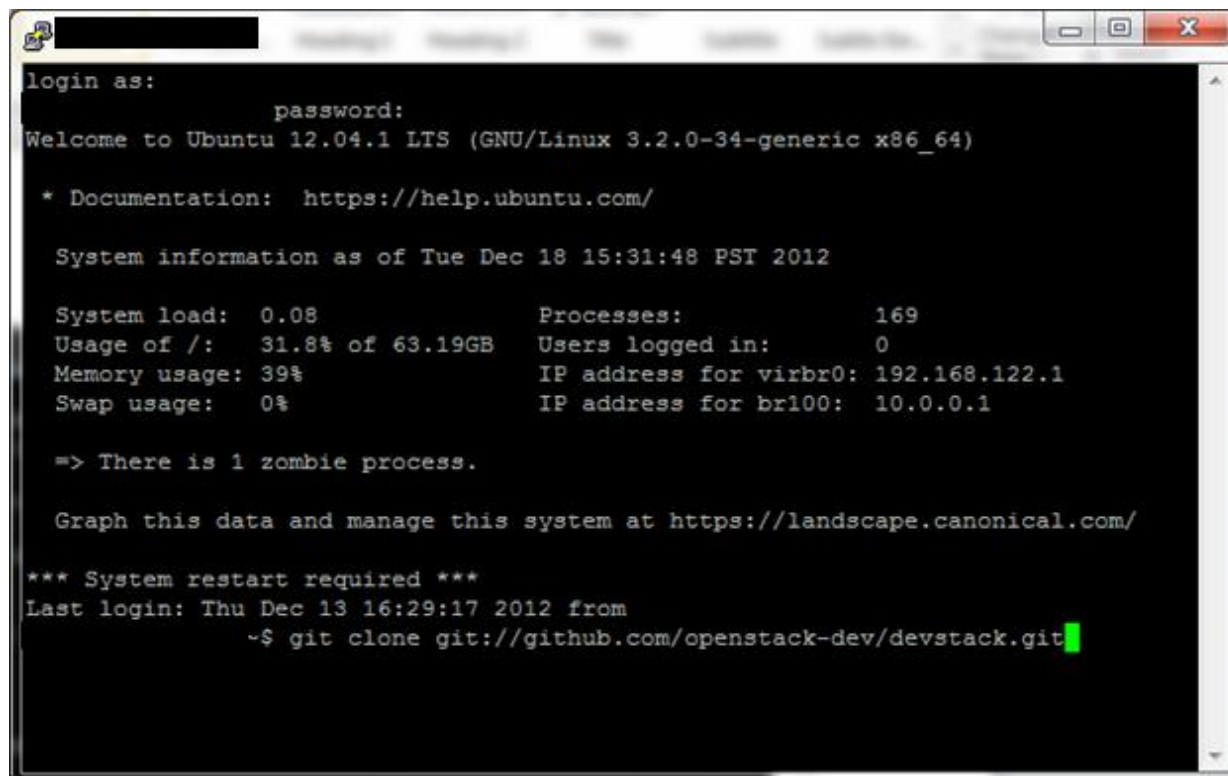
First, we have to connect to the server via a Secure Shell (SSH). Therefore we are using the assigned hostname “bus-mac16” with the standard TCP port 22. One of the free tools is PuTTY, which is an open source SSH and telnet client. The latest version can be downloaded at: <http://www.putty.org>



5.2) Downloading devstack

After the connection is established, we login on the shell and download devstack to the server using the following command:

```
“git clone https://github.com/Openstack-dev/devstack.git”4
```

A terminal window showing the login process on an Ubuntu 12.04.1 LTS system. The user enters a password and is greeted with system information including system load, memory usage, and IP addresses. The terminal also shows a message about a zombie process and a system restart requirement. The command to clone the devstack repository is entered at the prompt.

```
login as:
          password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-34-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Tue Dec 18 15:31:48 PST 2012

System load:  0.08          Processes:            169
Usage of /:   31.8% of 63.19GB Users logged in:      0
Memory usage: 39%          IP address for virbr0: 192.168.122.1
Swap usage:   0%           IP address for br100:  10.0.0.1

=> There is 1 zombie process.

Graph this data and manage this system at https://landscape.canonical.com/

*** System restart required ***
Last login: Thu Dec 13 16:29:17 2012 from
          ~$ git clone git://github.com/openstack-dev/devstack.git
```

⁴ <http://devstack.org/>

5.3) Execution of the script “stack.sh”

Afterwards we switch to the “/devstack” directory and execute the “stack.sh” script by using the “screen” program “./”

```
~/devstack
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$
~/devstack$ ls
AUTHORS      exercise.sh  lib          README.md    stack-screenrc  unstack.sh
eucarc       files       LICENSE     rejoin-stack.sh  stack.sh
exerciserc   functions   localrc     samples      tests
exercises    HACKING.rst openrc      stackrc      tools
--          --          --          --            --
~/devstack$ ./stack.sh
```

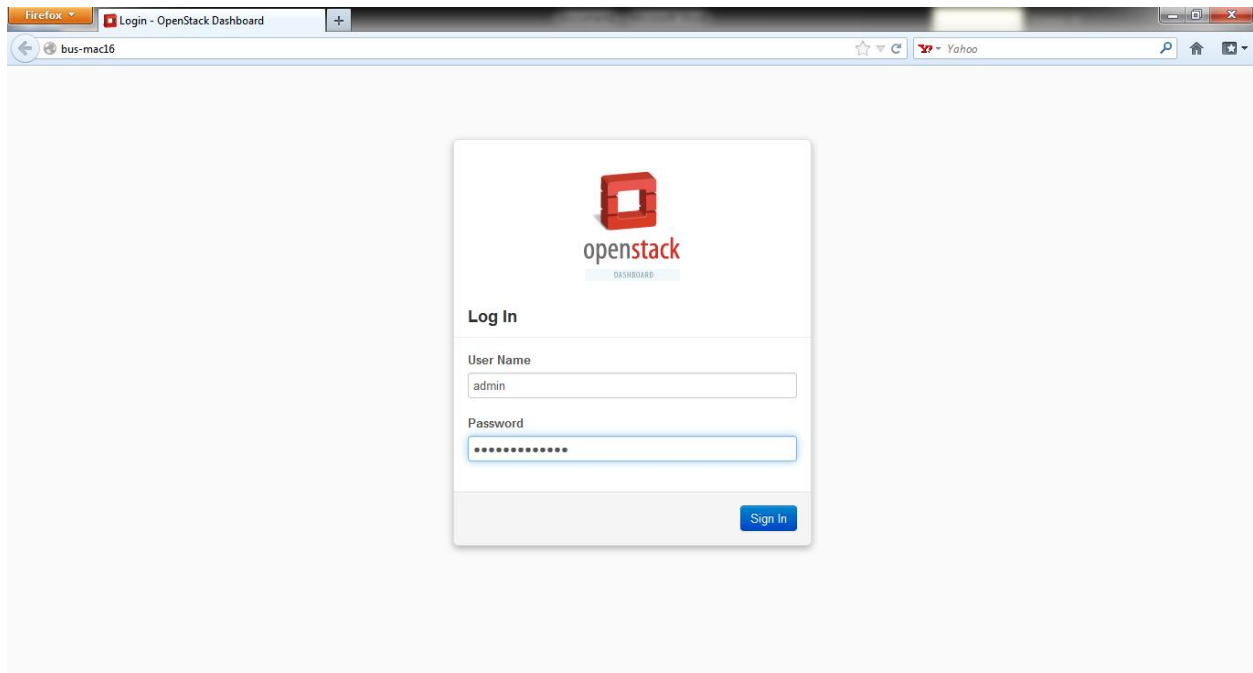
After the script was successfully executed, it shows the configuration and under which IP the system is reachable.

```
~/devstack
+ is_service_enabled tempest
+ services=tempest
+ for service in '${services}'
+ [[ ,g-api,g-reg,key,n-api,n-ort,n-obj,n-cpu,n-net,n-cond,cinder,c-sch,c-api,c-vol,n-sch,n-novnc,n-xvnc,n-cauth,horizon,rabbit,mysql, == ,tempest, ]]
+ [[ tempest == \n\o\v\l ]]
+ [[ tempest == \c\i\n\d\e\r ]]
+ [[ tempest == \c\o\l\l\o\m\l\c\l\o\r ]]
+ [[ tempest == \q\l\l\l\l\c\l\e ]]
+ [[ tempest == \q\l\l\l\l\c\l\m ]]
+ return 1
+ [[ -x /home/sverma/devstack/local.sh ]]
+ set +o xtrace

Horizon is now available at http://130.212.
Keystone is serving at http://130.212      5000/v2.0/
Examples on using novaclient command line is in exercise.sh
The default users are: admin and demo
The password: ██████████
This is your host ip: 130.212
stack.sh completed in 150 seconds.
~/devstack$
```

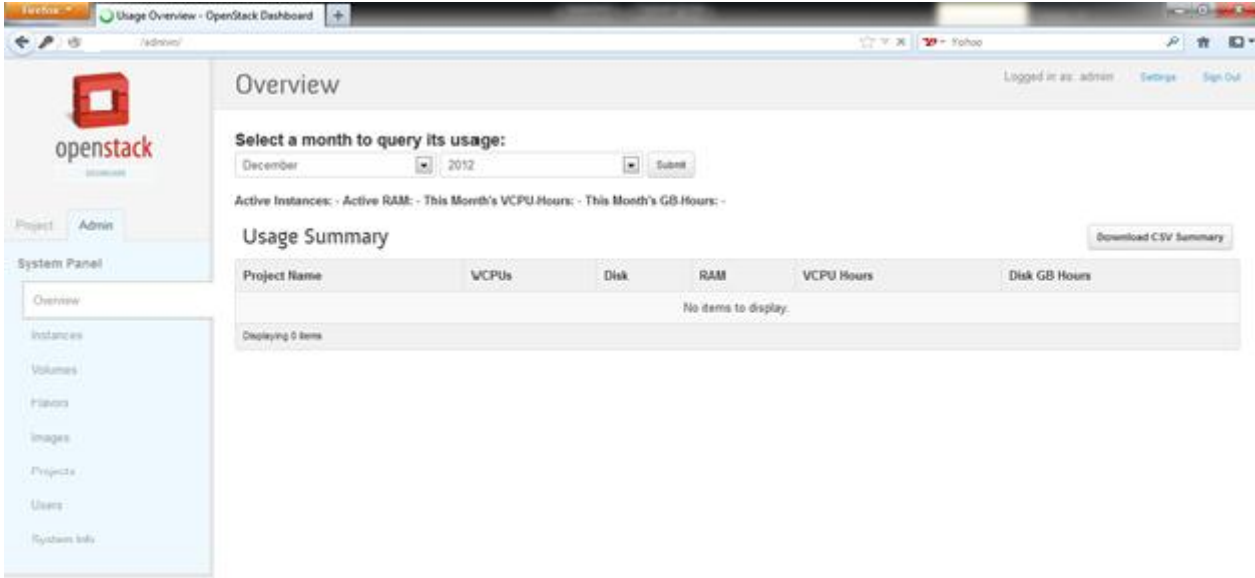

5.4) Log In to Openstack

From now on, the system is also available under the GUI with using any web browser. It's possible to either connect to the IP address or the DNS name. As you can see in the following screenshot we used the DNS name because it's easier to remember. Horizon manages the GUI for Openstack.



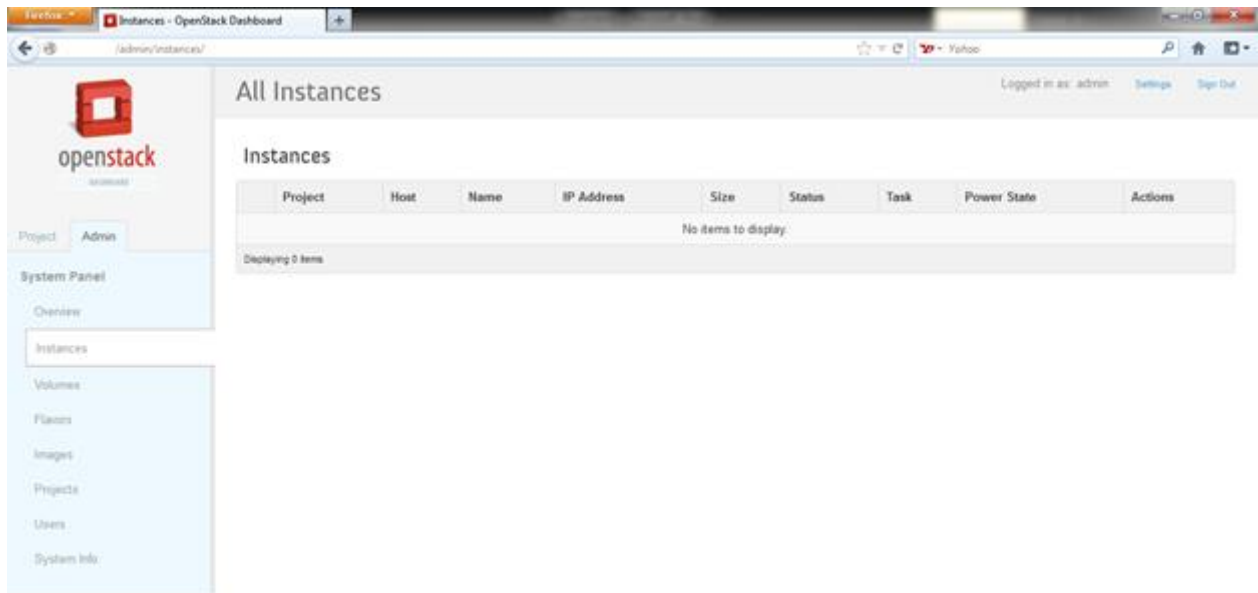
5.5) Overview

The Overview page is the default page after you log into Openstack. There are many options on the left hand side to choose from, the Overview screen allows users to see which projects are currently running, along with details about the projects.



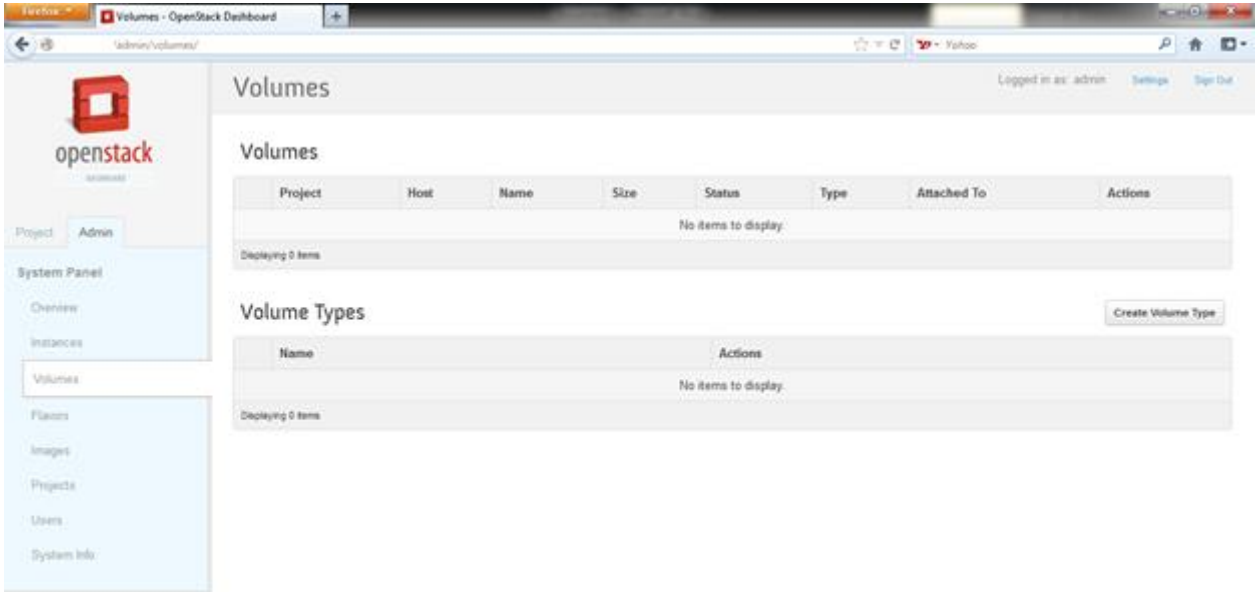
5.6) Instances

The Instances page is next, it shows which instances are running, as well as the IP address, Size, Status, Task, Power State and management options for each instance. Instances are individual VM's that are running on physical compute nodes. Nova handles the instances for Openstack.



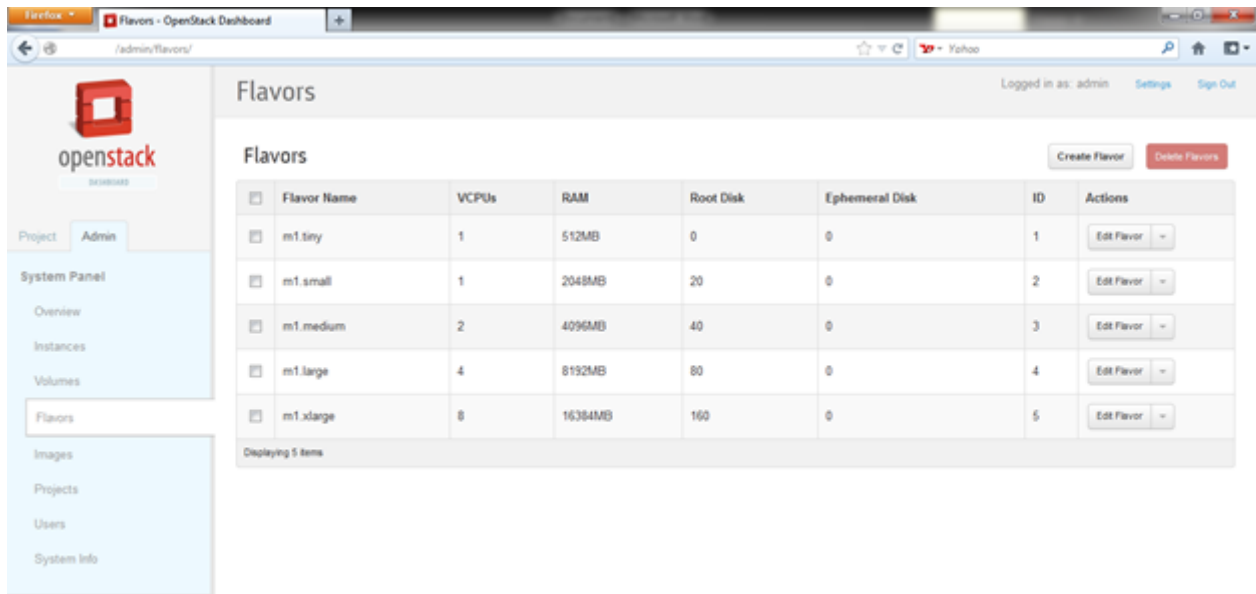
5.7) Volumes

In this installation, Cinder is installed instead of Swift. Cinder is for block storage. On the Volumes page, Openstack shows which volumes are active at the moment. We are able to create different volume types with the Create Volume Type button.



5.8) Flavors

Flavors is the next page. Flavors are customizable settings to create each virtual machine with. The created machines will have the specs of the flavor. The configurable options for flavors are VCPUs, RAM, Root Disk, and Ephemeral Disk. Ephemeral disks are local disk devices. This makes it easy to customize each VM to meet performance needs.



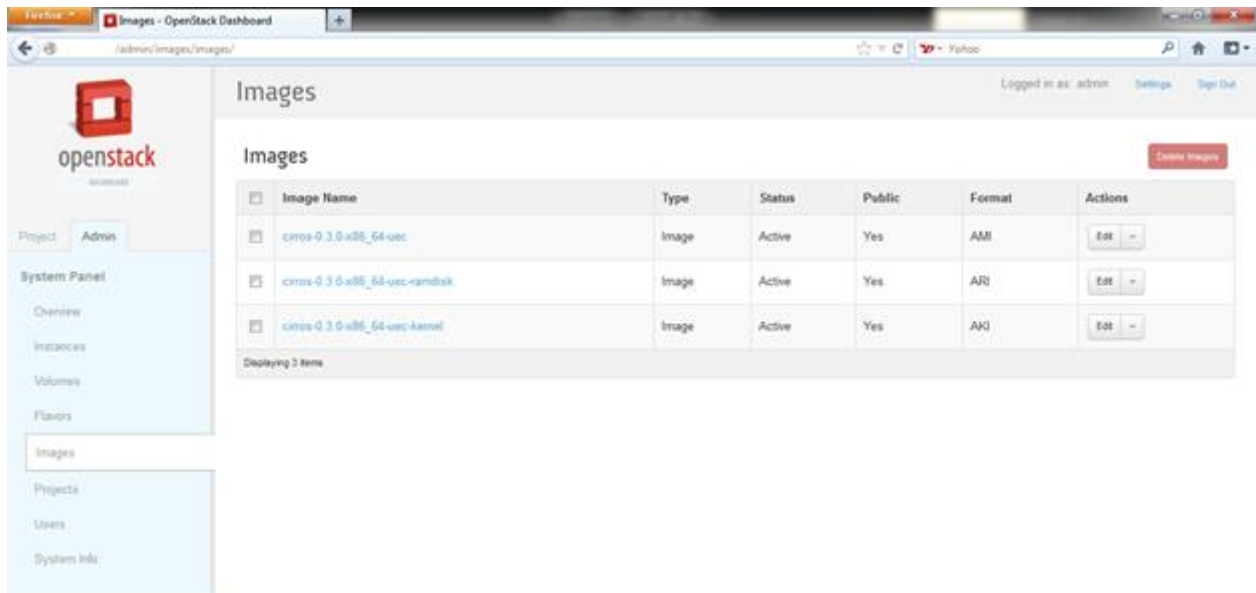
The screenshot shows the OpenStack Dashboard interface for managing flavors. The page title is "Flavors" and the user is logged in as "admin". The dashboard includes a sidebar with navigation options: Project, Admin, System Panel, Overview, Instances, Volumes, Flavors (selected), Images, Projects, Users, and System Info. The main content area displays a table of flavors with the following data:

Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	ID	Actions
m1.tiny	1	512MB	0	0	1	Edit Flavor
m1.small	1	2048MB	20	0	2	Edit Flavor
m1.medium	2	4096MB	40	0	3	Edit Flavor
m1.large	4	8192MB	80	0	4	Edit Flavor
m1.xlarge	8	16384MB	160	0	5	Edit Flavor

Buttons for "Create Flavor" and "Delete Flavors" are located at the top right of the table. The footer of the table indicates "Displaying 5 items".

5.9) Images

Images is the next page. This is where users can choose specific images for Openstack to deploy. The default image is cirros 0.3.0 x86_64. Images can be easily added by editing the stackrc text tile in the command line. Glance is responsible for storage and management of the images.



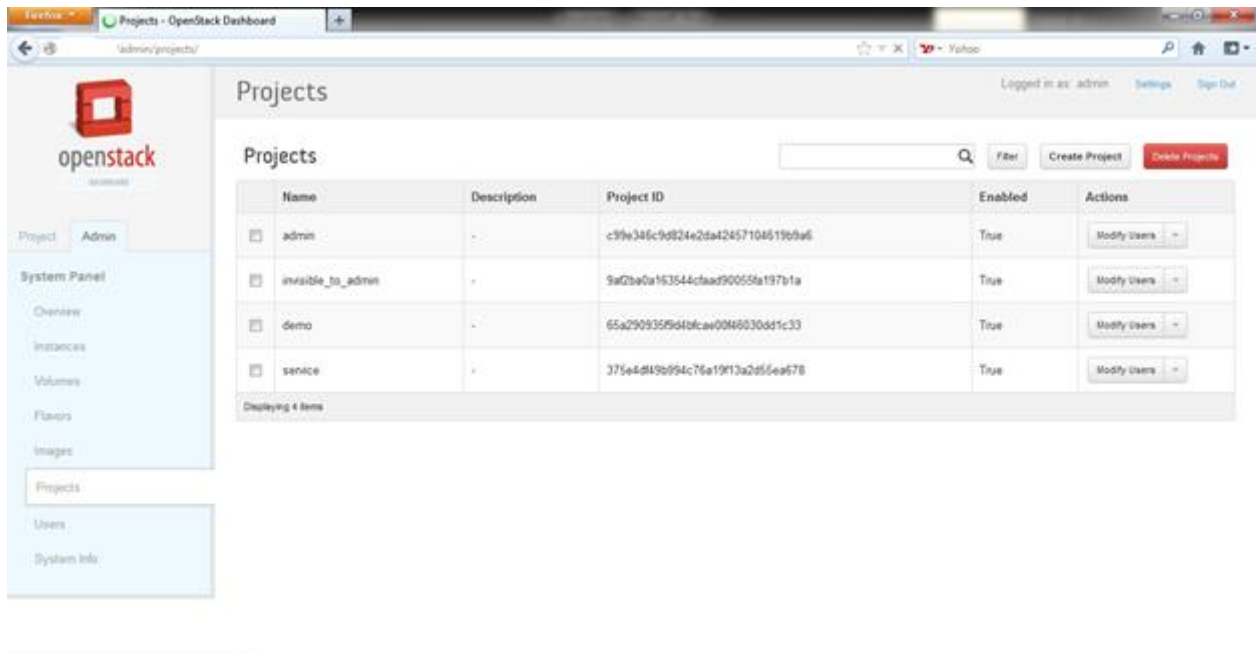
The screenshot shows the OpenStack Images dashboard in a web browser. The page title is "Images" and the user is logged in as "admin". The dashboard displays a table of images with the following columns: Image Name, Type, Status, Public, Format, and Actions. There are three images listed:

<input type="checkbox"/>	Image Name	Type	Status	Public	Format	Actions
<input type="checkbox"/>	cirros-0.3.0-x86_64-uec	Image	Active	Yes	AMI	Edit -
<input type="checkbox"/>	cirros-0.3.0-x86_64-uec-ramdisk	Image	Active	Yes	ARI	Edit -
<input type="checkbox"/>	cirros-0.3.0-x86_64-uec-kernel	Image	Active	Yes	AKI	Edit -

Below the table, it says "Displaying 3 items". On the left side, there is a navigation menu with the OpenStack logo and a "System Panel" section containing links for Overview, Instances, Volumes, Flavors, Images (selected), Projects, Users, and System Info. A "Create Images" button is visible in the top right corner of the dashboard area.

5.10) Projects

Projects are the next page. These and Users are basically the same thing, they were just different words for the different versions of Openstack. These are basically the user accounts. Keystone handles the management of users and projects.



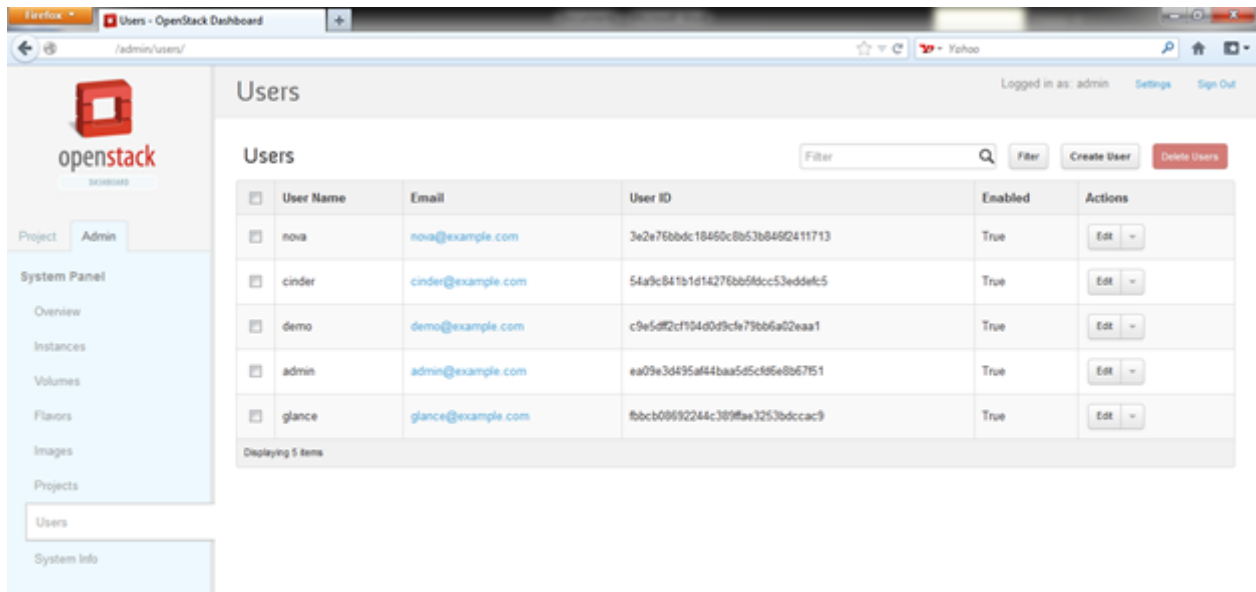
The screenshot shows the OpenStack Projects dashboard. The page title is "Projects" and the user is logged in as "admin". The dashboard displays a table of projects with the following data:

Name	Description	Project ID	Enabled	Actions
<input type="checkbox"/> admin	-	c99e346c9d24e2da1245710461969a6	True	Modify Users
<input type="checkbox"/> invisible_to_admin	-	9a2ba0a163544cfaad90055fa197b1a	True	Modify Users
<input type="checkbox"/> demo	-	65a2908359544bca0046030d41c33	True	Modify Users
<input type="checkbox"/> service	-	375e4d49b894c76a19f13a2b55ea678	True	Modify Users

Displaying 4 items

5.11) Users

Users are the next page. These and Projects are basically the same thing, they were just different words for the different versions of Openstack. These are basically the user accounts. Keystone handles the management of users and projects.



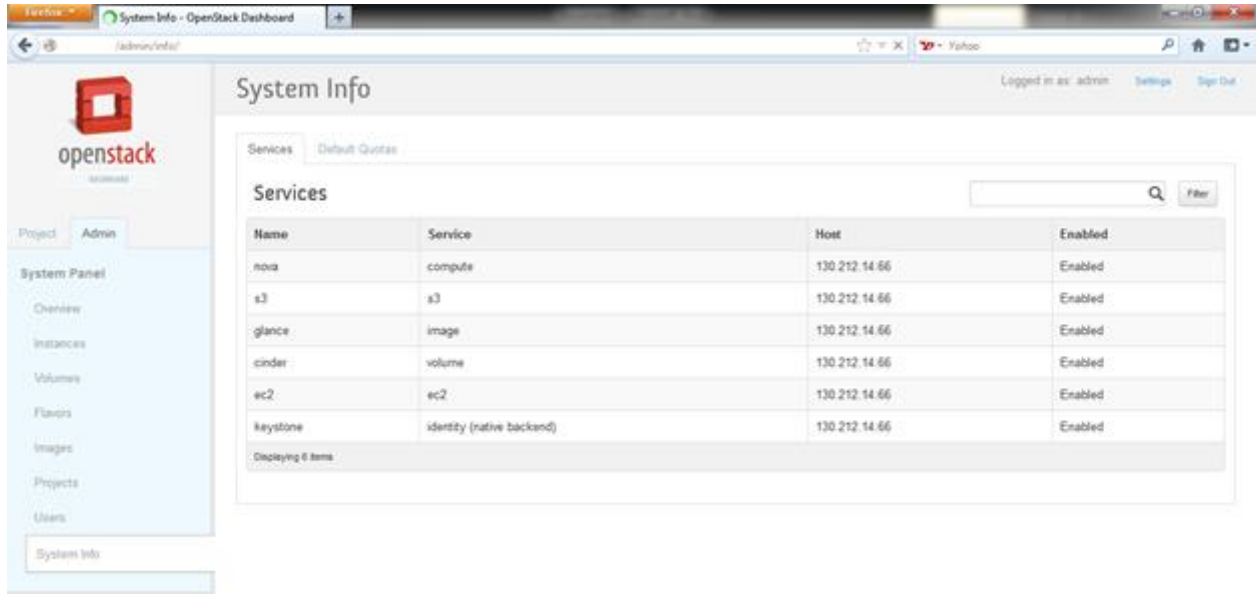
The screenshot shows the OpenStack Admin interface for managing users. The page title is "Users" and the user is logged in as "admin". The interface includes a sidebar with navigation options like "Project", "Admin", "System Panel", "Overview", "Instances", "Volumes", "Flavors", "Images", "Projects", "Users", and "System Info". The main content area displays a table of users with columns for "User Name", "Email", "User ID", "Enabled", and "Actions". There are also buttons for "Filter", "Create User", and "Delete Users".

<input type="checkbox"/>	User Name	Email	User ID	Enabled	Actions
<input type="checkbox"/>	nova	nova@example.com	3e2e76bbdc18460c8b53b8462411713	True	Edit
<input type="checkbox"/>	cinder	cinder@example.com	54a9c841b1d14276bb5d5cc53eddefc5	True	Edit
<input type="checkbox"/>	demo	demo@example.com	c9e5df2cf104d0d9cfe79bb6a02eaa1	True	Edit
<input type="checkbox"/>	admin	admin@example.com	ea09e3d495af44baa5d5cd9e8b67f51	True	Edit
<input type="checkbox"/>	glance	glance@example.com	fbcb08692244c389ffae3253bdccac9	True	Edit

Displaying 5 items

5.12) System Info

System Info is next, which just depicts which services are running, what the name of the services are, the host IP address and if they are enabled or disabled.



The screenshot shows the OpenStack System Info dashboard. The page title is "System Info" and the user is logged in as "admin". The dashboard displays a table of services with the following columns: Name, Service, Host, and Enabled. The table lists six services: nova, s3, glance, cinder, ec2, and keystone. All services are shown as "Enabled".

Name	Service	Host	Enabled
nova	compute	130.212.14.66	Enabled
s3	s3	130.212.14.66	Enabled
glance	image	130.212.14.66	Enabled
cinder	volume	130.212.14.66	Enabled
ec2	ec2	130.212.14.66	Enabled
keystone	identity (native backend)	130.212.14.66	Enabled

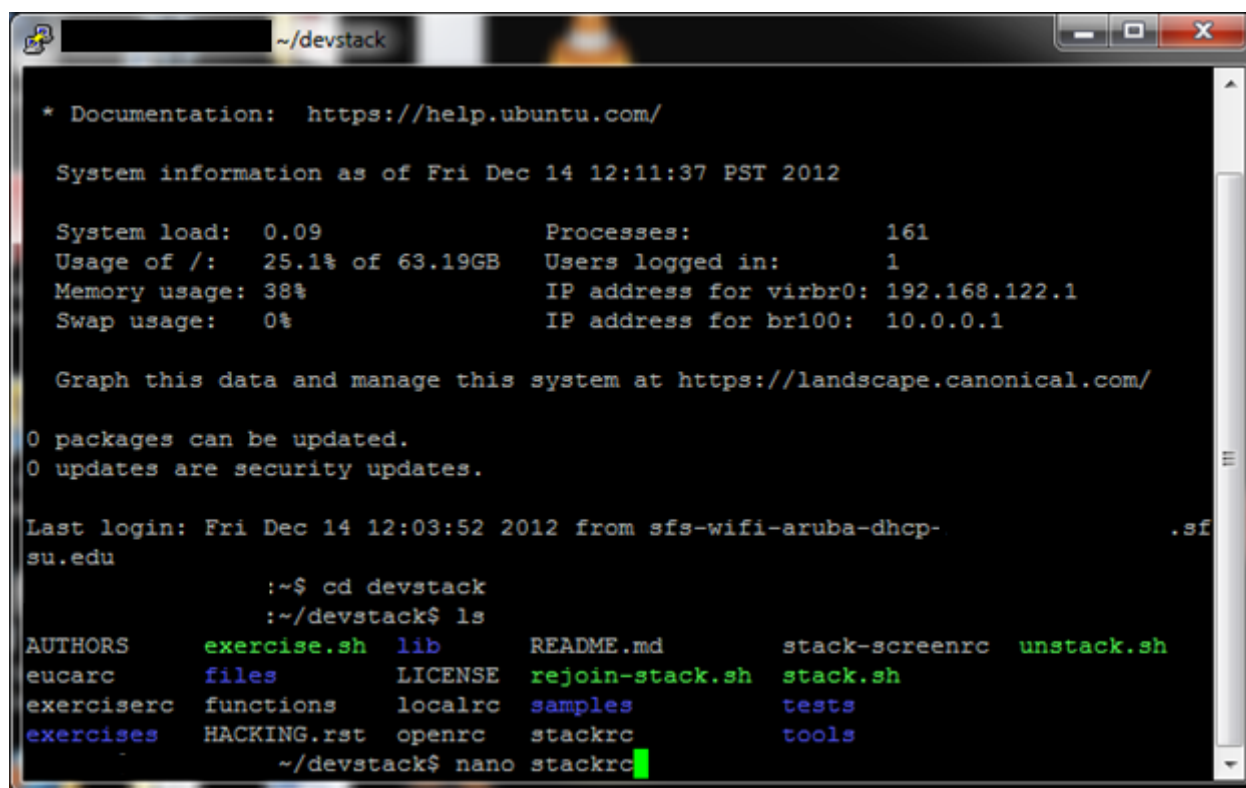
Displaying 6 items

5.13) Adding an image to Openstack

Open up the /devstack directory.

Use the command open up *stackrc* using the text editor *nano*.

“Nano stackrc”



```
~/devstack
* Documentation: https://help.ubuntu.com/

System information as of Fri Dec 14 12:11:37 PST 2012

System load:  0.09          Processes:           161
Usage of /:   25.1% of 63.19GB  Users logged in:    1
Memory usage: 38%          IP address for virbr0: 192.168.122.1
Swap usage:  0%            IP address for br100: 10.0.0.1

Graph this data and manage this system at https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

Last login: Fri Dec 14 12:03:52 2012 from sfs-wifi-aruba-dhcp-
su.edu

      :~$ cd devstack
      :~/devstack$ ls
AUTHORS  exercise.sh  lib          README.md    stack-screenrc  unstack.sh
eucarc   files        LICENSE      rejoin-stack.sh  stack.sh
exerciserc  functions   localrc     samples      tests
exercises HACKING.rst  openrc      stackrc      tools
      :~/devstack$ nano stackrc
```

This is the what the text file should look like.

```
GNU nano 2.2.6 File: stackrc
# glance as a disk image. If it ends in .gz, it is uncompressed first.
# example:
# http://cloud-images.ubuntu.com/releases/oneiric/release/ubuntu-11.10-server-cloudimg-armel-disk1.img
# http://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-rootfs.img.gz
# * OpenVZ image:
# OpenVZ uses its own format of image, and does not support UEC style images
#IMAGE_URLS="http://smoser.brickies.net/ubuntu/ttylinux-uec/ttylinux-uec-amd64-11.2.2.6.35-15.1.tar.gz" # old ttylinux
#IMAGE_URLS="http://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img" # cirros full disk image
#IMAGE_URLS="http://cloud-images.ubuntu.com/desktop/quantal/current/quantal-desktop-cloudimg-amd64.tar.gz"
#IMAGE_URLS="http://devel.trisquel.info/sugar/trisquel-sugar_3.0-LATEST_1686.iso"
#IMAGE_URLS="http://uec-images.ubuntu.com/desktop/lucid/current/lucid-desktop-cloudimg-amd64.tar.gz"

# Set default image based on `VIRT_DRIVER` and `LIBVIRT_TYPE`, either of
# which may be set in `localrc`. Also allow `DEFAULT_IMAGE_NAME` and
# `IMAGE_URLS` to be set directly in `localrc`.
case "$VIRT_DRIVER" in
  openvz)
    DEFAULT_IMAGE_NAME=$(DEFAULT_IMAGE_NAME:-ubuntu-11.10-x86_64)
  *)
    ;;
esac
```

Scroll all the way down, delete the # in front of the image you want to use. You can also add additional images by separating each image with a comma.

For example:

```
GNU nano 2.2.6 File: stackrc
# http://cloud-images.ubuntu.com/releases/oneiric/release/ubuntu-11.10-server-cloudimg-armel-disk1.img
# http://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-rootfs.img.gz
# * OpenVZ image:
# OpenVZ uses its own format of image, and does not support UEC style images
#IMAGE_URLS="http://smoser.brickies.net/ubuntu/ttylinux-uec/ttylinux-uec-amd64-11.2.2.6.35-15.1.tar.gz" # old ttylinux-uec image
#IMAGE_URLS="http://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img" # cirros full disk image
#IMAGE_URLS="http://cloud-images.ubuntu.com/desktop/quantal/current/quantal-desktop-cloudimg-amd64.tar.gz"
#IMAGE_URLS="http://devel.trisquel.info/sugar/trisquel-sugar_3.0-LATEST_1686.iso"
#IMAGE_URLS="http://uec-images.ubuntu.com/desktop/lucid/current/lucid-desktop-cloudimg-amd64.tar.gz,http://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-rootfs.img.gz"

# Set default image based on `VIRT_DRIVER` and `LIBVIRT_TYPE`, either of
# which may be set in `localrc`. Also allow `DEFAULT_IMAGE_NAME` and
# `IMAGE_URLS` to be set directly in `localrc`.
case "$VIRT_DRIVER" in
  openvz)
    DEFAULT_IMAGE_NAME=$(DEFAULT_IMAGE_NAME:-ubuntu-11.10-x86_64)
    IMAGE_URLS=$(IMAGE_URLS:-"http://download.openvz.org/template/precreated/ubuntu-11.10-x86_64.tar.gz");;
  libvirt)
    ;;
esac
```

Afterwards, reboot Openstack, and the image should appear under the Images page in the GUI.

